

Patent Application
Attorney Docket No. 07781.0080-00000
Customer No. 22,852

UNITED STATES PATENT APPLICATION

FOR

SYSTEMS AND METHODS FOR AUTOMATED DATA OBJECT PROCESSING

BY

FRANK WESTENDORF

AND

BENT FOSS PEDERSEN

DESCRIPTION OF THE INVENTION

Field of the Invention

[0001] The present invention generally relates to automated data object processing. More particularly, the invention relates to forming process chains of linked data objects in a data base of a computer system.

Background of the Invention

[0002] There are many cases and situations in which data objects contained in a data base are linked to each other so as to form process chains, i.e., the data objects linked to each other describe a development or the history of a process or an order of events. In this context, the term "process chain" does not only describe a mere one-dimensional sequence of data objects, but may also relate to higher-dimensional and complex chainings, networkings or meshings (i.e., a complex branch construct).

[0003] One example for data objects forming process chains of linked data objects, but not limited to, is in the field of calculation and remuneration of commission cases. Commission cases occur in the context of various business fields where employees or agents receive bonus or commission payments on the basis of achieved objects, such as for example in the field of insurance companies, where an employee or agent receives commission payments when he/she sells an insurance policy, etc. However, there are many other areas in sales and services where commission cases are handled. In situations with a complex environment and/or long-term contracts, a whole history of events (commission cases) develops

over time which leads to a situation with several data objects, each of which describes a commission case and which are based one on the other over time in a causal relation. In the most simple situation, this relation is a one-dimensional process chain of at least two data objects. As an example, Figure 1 illustrates a commission case chain with three commission cases forming a one-dimensional process chain with the oldest case (Case 1) depicted on the left hand side and the younger cases (Case 2 and Case 3) placed subsequently along a time axis showing to the right hand side of Figure 1. Arrows between the respective cases indicate the succession in time.

[0004] As can be seen from Figure 1, each commission case contains respective sub-objects which comprise information of what has been done in the commission case. As sub-objects are independent of the case, they play an important rule when a chain in commission cases has to be updated. The situation illustrated in Figure 1 could, for example, stand for a life insurance contract which was signed at a given date (Case 1). This led to a commission payment to the employee who sold that life insurance. At a later date, the amount covered by this life insurance was raised which led to a further commission payment to the relevant employee (who could be the same employee as in Case 1 or another employee, e.g., a successor to the employee who sold the insurance in the first place). Again later, the amount was raised again (or any other change to the insurance contract took place) resulting in a third commission payment.

[0005] Other, more complex situations may exist. For example, in the case of health insurance, there could be the case of two single persons, each with his or

her own health insurance contract, and which might comprise additional benefit agreements (e.g., getting married and having children with contracts getting merged, cancelled, creating subcontracts and eventually new contracts emerging from subcontracts when the children are grown up). Obviously, there are (basically) no limits to complexity.

[0006] In a situation where a commission case has to be updated, the whole chain of commission cases has to be updated, too, as remuneration amounts can influence other remuneration amounts as the remuneration amounts depend on the remuneration they correct/influence. Cases with such remunerations have to be considered when a chain of commission cases is updated. In such a case, sub-objects play an important role as they are independent of the case. However, in the systems known from the prior art, it is only possible to update the youngest of the commission cases, i.e., the commission case last posted in the commission case chain.

SUMMARY OF THE INVENTION

[0007] In accordance with an aspect of the invention, a data object processing system is provided that allows for fast and reliable updating of process chain data objects, even in very complex process chain branch structures. In accordance with another aspect of the invention, a data object processing system is provided in which the data objects are consistent after processing of an update.

[0008] According to one embodiment, a computing unit of a computer system comprises a selection module, a preparation module, and a process module. In addition, a data base is provided that contains data objects forming process

chains of linked data objects. The selection module may be adapted to allow for the selection of a data object among the data objects contained in the data base. On the basis of a selected data object, the preparation module may identify at least one process chain containing the selected data object. The preparation module may also determine all of the data objects of the data base which are linked with the selected data object in causal relation. Further, the process module may execute the desired processing of the selected data object and the data object(s) being in causal relation with the selected data object.

[0009] In another embodiment, a method is provided that comprises, by way of a non-limiting example, the following steps: identifying, after a data object to be processed has been selected, at least one process chain containing the selected data object to be processed; determining all of the data objects which are linked with the selected data object in causal relation; and processing the determined data objects and the initial data object.

[0010] The step of processing may comprise subsequently resetting or cancelling one determined data object after the other and consecutively recalculating, if necessary (i.e., if the determined data object was not cancelled), the determined data objects reset in the resetting step being in causal relation with the selected data object.

[0011] In one embodiment, the processing steps are configurable by switches. For example, a user could define that resetting and recalculation of a determined data object is replaced by cancellation.

[0012] As exemplary uses of the invention, it is possible to handle automatically deep and complex networks of data objects, such as commission cases, within one single process to resolve pending case queues or to recalculate them in changed context.

[0013] Further, the data objects may contain a time index, and the resetting of the determined data objects (as executed by the process module) may start with the data object containing the most resent time index and step back consecutively to the selected data object.

[0014] In another embodiment, a computer program is provided with program coding means which are suitable for carrying out a process consistent with the invention, such as described above, when the computer program is run on a computer. The computer program may be stored on a computer-readable medium.

[0015] Further features and embodiments of the invention will become apparent from the description and the accompanying drawings.

[0016] It will be understood that the features mentioned above and those described hereinafter can be used not only in the combination specified, but also in other combinations or on their own, without departing from the scope of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] The accompanying drawings provide a further understanding of the invention and are incorporated in and constitute a part of this specification. The drawings illustrate various embodiments of the invention, and, together with the description, serve to explain the principles of the invention.

- [0018] In the drawings,
- [0019] Figure 1 is a schematic illustration of a conventional data object chain with three data objects, the data objects describing commission cases;
- [0020] Figure 2 is a schematic view of an exemplary data object processing computer system, according to an embodiment of the invention;
- [0021] Figure 3 is a schematic block diagram depicting three exemplary modules of the computer system according to the embodiment of Figure 2;
- [0022] Figure 4 illustrates, by means of a block diagram, an exemplary processing of linked data object process chains (case chains), according to an embodiment the invention;
- [0023] Figure 5 shows an exemplary table identifying a sequence of case processing;
- [0024] Figure 6 is a schematic illustration of exemplary processing situations;
- [0025] Figure 7 is a schematic block diagram representation illustrating exemplary sort order processing; and
- [0026] Figure 8 is a representation according to Figure 7 with additional exemplary sorting indications.

DESCRIPTION OF THE EMBODIMENTS

- [0027] Figure 2 shows a schematic block diagram representation of an exemplary data object processing computer system 10, according to an embodiment of the present invention. Computer system 10 may comprise a server or computing unit 12 connected with at least one input/output means 18. A software solution may

be installed on computing unit 12 comprising three modules, namely a selection module 16, a preparation module 14 and a process module 22. The computing unit 12 may be connected to a data base 20 on which a plurality of data objects is stored.

[0028] Figure 3 shows an exemplary succession in operation of three modules; that is, selection module 16, preparation module 14 and process module 22 described above.

[0029] For purposes of illustration, embodiments of the invention will now be described in more detail in connection with commission cases. It is to be understood that embodiments of the invention can be employed in any other field in which subsequent data objects are linked to each other in a causal relation and influence there contents and/or results. In this context, the term "report" is to be understood as a synonym for "computer program".

[0030] According to one embodiment of the invention, selection module 16 may select one or more commission cases that are to be processed, preparation module 14 may prepare the cases by finding additional cases, grouping and sorting them, and process module 22 may process the cases determined before.

[0031] Selection module 16 may be a module created depending on the context of its use. The selection module 16 could be a report that is based on some selection criteria that select the cases or it could be a report with a user interface that allows the user to select cases based on certain selection criteria, such as a case ID or all cases posted in a certain period of time, or the like. The selected cases may then be transferred to the preparation module 14.

[0032] The preparation module 14 may require a selected list of commission cases as input that is provided by the selection module 16. By way of an example, the list of selected cases (selected data objects) is referred to as the list of initial cases (or initial data objects). The preparation step performed in the preparation module 14 may need to locate additional required cases that need to be processed before the initial selected cases can be processed. For example, in the simple case illustrated in Figure 1, if case 1 is selected as the initial case, the preparation module 14 would determine that cases 3 and 2 are to be processed before case 1 can be processed.

[0033] In one embodiment, the cases determined by the preparation module 14 may be grouped and sorted. The result of this grouping and sorting could be groups of sorted cases that can later be processed independently. As a prerequisite for the parallel processing, the groups may be disjoint.

[0034] The result of the preparation in preparation module 14 may be a task list based on the grouped and sorted commission cases which contain a list of instructions on how to handle the cases and which may form the basis for the process module 22 to process the determined commission cases. By way of a non-limiting example, the arrangement within a group is such that a risk-free and consistent processing by the process module according to the application logics is guaranteed. Further, the group may be closed, and if an interruption occurs during processing it can be resumed at the commission case where the interruption occurred.

[0035] Also, the preparation module 22 may provide program interfaces to maintain the initial list of commission cases and to start the preparation.

[0036] Finally, the process module 22 may process the task list generated and provided by the preparation module 14. In one embodiment, it is possible to start several processing tasks in parallel, each process processing one task list.

[0037] The preparation module 14 and/or the process module 22 may each comprise at least one interface. Interfaces for the preparation module 14 are, for example: (i) adding a commission case to the list of initial selected commission cases that have to be processed; and (ii) starting the preparing process (e.g., creating the groups, sorting the cases and creating the task lists). Interfaces for the processing module 22 comprise, for example: (i) maintaining the task list and a list with overall information about the task list; and (ii) controlling a possible parallel process.

[0038] In one embodiment, when a commission case is processed, both subsequently posted commission cases in the commission case chain, cases that contain rebundled subobjects and commission cases with correction remunerations, can be taken into account.

[0039] Referring now to the exemplary embodiment of Figure 4, a commission case chain 30 consisting of four commission cases 1 to 4 is shown. As can be seen from the time axis shown on the lower right hand side of Figure 4, case 1 is the oldest case in commission case chain 30 and case 4 is the youngest case.

[0040] Two further exemplary commission case chains 32 and 34 are shown in Figure 4. Commission case chain 32 comprising cases Y1 to Y3 and

commission case chain 34 comprising cases Z1 to Z3. Case Y1 of commission case chain 32 is older than case 1 of commission case chain 30 and case Y2 of commission case chain 32 is younger than case 1 of commission case chain 30. Similarly, case Z1 of commission case chain 34 is older than case 3 of commission case chain 30 and case Z2 of commission case chain 34 is younger than case 3 of commission case chain 30.

[0041] Exemplary case 3 contains a sub-object S3 that is rebundled to case Z2 (i.e., case 3 and case Z2 are linked in causal relation). As case Z2 is younger than case 3, this means that case Z2 has to be considered if any of the cases 1 to 3 is selected as an initial case and has to be handled/processed because the valuation for the sub-object S3 could change in case 3 and as a result, the valuations in the cases Z2 and Z3 could change, too. The valuation for an object may contain the history of the case chain development and, thus, may produce a concatenation of the case statuses over the time. The valuation may also serve as basis for the renumeration.

[0042] Exemplary case Y2 contains a sub-object SY2 that is rebundled to case 2 (i.e., cases Y2 and 2 are linked in causal relation). However, in this example, this rebundling is of no importance as the valuations in case Y2 (which is older than case 2) may not change as a result of an update/processing of case 2.

[0043] Figure 5 shows a exemplary table containing a list of exemplary tasks which may be processed depending on which one of the cases 1 to 4 of commission case chain 30 is selected as an initial case.

[0044] By way of an example, the first row identifies the most simple occurrence, i.e., a selection of case 4. As case 4 is the youngest case in commission case chain 30 and not rebundled or linked to any other case, the processing of case 4 comprises parking or locking of case 4 and resolving it.

[0045] Should case 3 be selected as initial case, case 4 is parked first and then cases Z3 and Z2 are parked as sub-object S3 is rebundled from case 3 to case Z2. Afterwards, the cases 3, Z2, Z3 and 4 are resolved. Regarding the terminology used herein, "park" is an example for "reset" as defined above and "resolve" is an example for "recalculate" as defined above. Consistent with the invention, a parked case may represent a case that is placed on hold or pending. A parked case may not be treated, particularly there may be no simulation, until it is resolved. A case may be reset by taking the case out of the commission case for at least a period of time (roll back a case that is not an initial case) for the purpose of keeping it out of the line so that no changes or recalculations can occur. A reset (or parked) case can, when it is resolved, either be set to an initial state, cancelled or recalculated.

[0046] If case 2 is selected as initial case, cases 3 and 2 are parked first. Then, case 2 is resolved, and then case 3 is resolved as described above.

[0047] If case 1 is selected as initial case, case 2 has to be parked first as described above. Then case 1 is resolved, and then case 2 is resolved as described above.

[0048] By way of an example, if there is a remuneration amount in the current case that is corrected by a remuneration amount calculated in another case, that case has to be considered, too. The reason is that if the remuneration in the

current commission case changes then the correction remuneration might change, too, and therefore the commission case has to be taken into account.

[0049] Figure 6 shows four exemplary situations of remuneration corrections, commission case 2 being the case examined. By way of non-limiting examples, the situations indicated with dashed arrows and reference numerals ① and ② are both situations where a remuneration in case 2 corrects/influences a remuneration from another commission case, i.e., case 1 in situation ① and case Y2 in situation ②. These situations do not have to be dealt with because the results of cases 1 and Y2 do not change when case 2 changes. The situations depicted by dashed arrows with reference numerals ③ and ④ indicate situations where a remuneration amount is corrected by a remuneration amount that is part of another commission case, i.e., cases 3 in situation ④ and case Z1 in situation ③. If the remuneration amount in commission case 2 changes, then the correcting remuneration amount could change too. Therefore, these cases have to be dealt with in the same task list and they have to be located/determined during the preparation in preparation module 14.

[0050] As described above, the process according to embodiments of the invention may comprise the three modules, including selection module 16, preparation module 14 and process module 22. Selection module 16 provides preparation module 14 with the one or more selected cases. After the preparation module 14 has determined additional cases, grouped and sorted the cases and

generated the task list, the task list may be provided to process module 22 and may be processed by the process module.

[0051] By way of example, preparation module 14 may receive a list of initial commission cases provided by selection module 16, which list may contain the commission cases that need to be processed. The preparation module 14 may then perform the following steps of locating/determining and grouping additional cases, sorting cases inside a group and generating a task list. The step of locating may consist of locating additional commission cases that need to be processed before the initial selected case(s) can be processed. All additional commission cases may be placed in the same group as the initial case. The commission cases may then be sorted into the right order within a given group in order to ensure that the cases are processed in the correct order.

[0052] A exemplary sorting step is illustrated in the exemplary embodiment of Figure 7. In this example, Figure 7 again shows commission case chains 30 and 34 of Figure 4, exemplary case 3 of commission case chain 30 being linked to exemplary case Z2 of commission case chain 34 due to the rebundled sub-object S3. The exemplary situation shown in Figure 7 corresponds to exemplary case 1 of commission case chain 30 being selected as initial case. In the first step of preparation module 14, the additional exemplary cases 2, 3, 4, Z2 and Z3 were determined as cases to be processed. Exemplary case Z1 not having to be processed as it is older than case 3 and thus older than the linkage between case 3 and case Z2, and there is no influence from a change of case 3 or case Z2 on case Z1 (cf Figure 5).

[0053] In the next step the cases may be sorted in an order in which they can be processed, i.e., first parked and then resolved. An exemplary sort order is indicated in Figure 7 by means of numbers 1 to 6 in squares. As can be seen from Figure 7, case Z3 is the first case parked, then case 4 is parked, followed by case Z2 and cases 3 to 1. This may correspond to the "age" of each case, the age of a case being advantageously determined by a time index contained in each data object. Thus, the sort order indicated in Figure 7 may start with the data object containing the most recent time index (here: case Z3) and then step back consecutively to the selected data object case 1. When resolving or recalculating the cases, the given sort order may be processed in reverse order, i.e., case 1 is processed first, followed by cases 2 and 3, then case Z2, followed by cases 4 and Z3.

[0054] Alternatively, another sort order could start with the most recent, i.e., youngest case in the direct line of the selected case which would be case 4 in the situation shown in Figure 7, followed by the subsequent order cases in the same line except such an older subsequent case would be linked to another line. This is the situation for exemplary case 3 in Figure 7 being linked to case Z2 of the line of commission case chain 34. Therefore, case 3 is not the next case in the sort order but the most recent case in the parallel commission case chain 34 which is case Z3, followed by all the subsequent older cases until the case is reached which is linked to the previous line or chain. Then the sort order is continued in the initial line, i.e., with cases 3 to 1. This sort order is given in row 4 of Figure 5. Obviously, sorting is illustrated by means of rather low complex process chains, but it should be obvious

that the principles laid down above according to the invention can be applied to much more complex data object chain networks with a plurality of links due to rebundled subobjects.

[0055] As a next step, the task list to be provided to the process module 22 may be generated in preparation module 14 based on the grouped and sorted list of commission cases. An exemplary complete task list is illustrated in Figure 8, the numbers depicted in Figure 8 giving the previously described order of parking and resolving the cases.

[0056] In the example of Figure 8, exemplary case Z3 is parked first, then cases 4, Z2, 3, 2 and finally case 1. Case 1 is resolved and then the rest of the cases are resolved in the reverse order in which they were parked.

[0057] The order of the parking may be the same as the order of the sorted cases of Figure 7. The commission cases may be added to the task list and signed the action *park*, except for the initial case which is signed the action *park initial*. The cases may then be copied in the reverse order to the task list and at the same time assigned the action *resolve*, except for the initial case which may be signed the action *resolve initial*.

[0058] Thus, it is possible to handle very large amounts of data. Exemplary embodiments of the invention may differentiate multiple processing types, support parallel administration of processes and/or protect mass processing against each other and against single processing.

[0059] Obviously, more than one data object may be selected as initial case and therefore, a grouped list can comprise more than one initial case which also

applies to the task lists so that the actions *park initial* and *resolve initial* can occur more than once in a task list.

[0060] The process module 22 may then processes the cases as specified in the task list. If more than one task list exists, each disjoined, it is possible to process two or more task lists in parallel.

[0061] Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope and spirit of the invention being indicated by the following claims.